IN THE CLAIMS:

Please cancel claims 5 and 15. **Please also amend** claims 1, 6-9, 11, 12, 14, and 16-20 as shown in the complete list of claims that is presented below.

1. (currently amended) A method of initializing a computer system equipped with a debugging system, wherein the computer system has a CPU, [[a]] local, peripheral and expansion ~~bus,~~ buses, [[a]] first and second ~~bridge~~ bridges, and a ROM coupled to the expansion bus and storing a first BIOS code, and wherein the debugging system is coupled to the peripheral bus, the method comprising the steps of:

operating the CPU in a normal mode wherein first data requests directed to the ROM are routed to the local bus by the CPU;

operating the CPU in a debugging mode wherein second data requests directed to the debugging system are routed to the local bus by the CPU;

transferring one of the data requests from the local bus to the peripheral bus via the first bridge;

responding via the second bridge to the first data requests on the peripheral bus ~~with~~ so that the first BIOS code stored in the ROM ~~to be~~ is loaded in the CPU; and

responding via the debugging system to the second data requests on the peripheral bus ~~with the~~ so that a second BIOS code stored ~~therein to be~~ in the debugging system is loaded in the CPU[[.]],

wherein the CPU is switched to the debugging mode if responding via the second bridge to the first data requests fails, and the first BIOS code is read or reprogrammed by the debugging system for debugging.

2. (original) The method as claimed in claim 1, wherein the second BIOS code is programmed by the debugging system.

3. (original) The method as claimed in claim 2, wherein the debugging system comprises:

an interface card coupled to the peripheral bus; and

a second computer system coupled to the interface card.

4. (original) The method as claimed in claim 1 further comprising the step of:

when the CPU is switched to the debugging mode, retrieving and displaying contents of registers in the CPU via the debugging system.

Claim 5 (cancelled).

6. (currently amended) The method as claimed in claim 1 further comprising the step of:

when the CPU is switched to the debugging mode, overwriting the first BIOS code in the ROM with the second BIOS code by the debugging system through the second bridge.

7. (currently amended) The method as claimed in claim 1, wherein switching between the normal and debugging ~~mode~~ modes is performed by enabling and disabling an A20 gate of the CPU respectively.

8. (currently amended) The method as claimed in claim 1, wherein the peripheral and expansion ~~bus~~ buses are a PCI bus and an ISA bus, respectively, and the first and second bridges are a north bridge and a south bridge, respectively.

9. (currently amended) The method as claimed in claim 1, wherein the second bridge responds to the first data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the first data requests and retrieving the first BIOS code in the ROM corresponding to the addresses.

10. (original) The method as claimed in claim 1, wherein the debugging system responds to the second data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the second data requests and retrieving the second BIOS code therein corresponding to the addresses.

11. (currently amended) A computer system capable of being initialized by a debugging system, comprising:

a CPU switched between a normal mode wherein first data requests are routed to ~~the~~ a local bus by the CPU and a debugging mode wherein second data requests directed to the debugging system are routed to the local bus by the CPU;

~~a local,~~ peripheral and expansion ~~bus,~~ buses in addition to the local bus, wherein the CPU routes one of the data requests to the local bus and the debugging system is coupled to the peripheral bus;

a ROM coupled to the expansion bus and storing a first BIOS code, to which the first data requests are directed;

a first bridge transferring one of the data requests from the local bus to the peripheral bus; and

a second bridge responding the first data requests on the peripheral bus ~~with~~ so that the first BIOS code in the ROM ~~to be~~ is loaded in the CPU[[;]],

wherein said debugging system responds the second data requests with a second BIOS code and loads the second BIOS code into the CPU[[.]], and

wherein the CPU is switched o the debugging mode if the second bridge fails to respond to the first data requests with the first BIOS code, and the debugging system reads or reprograms the first BIOS code for debugging.

12. (currently amended) The computer system as claimed in claim 11, wherein the second BIOS code ~~are~~ is programmed by the debugging system.

13. (original) The computer system as claimed in claim 12, wherein the debugging system comprises:

an interface card coupled to the peripheral bus; and

a second computer system coupled to the interface card.

14. (currently amended) The computer system as claimed in claim 11, wherein the debugging system retrieves and displays contents of registers in the CPU when the CPU is switched to the debugging mode.

Claim 15 (cancelled).

16. (currently amended) The computer system as claimed in claim 11, wherein the debugging system overwrites the first BIOS code in the ROM with the second BIOS code through the second bridge when the CPU is switched to the debugging mode.

17. (currently amended) The computer system as claimed in claim 11, wherein the CPU has an A20 gate and is switched between the normal mode and the debugging mode by enabling and disabling the A20 gate.

18. (currently amended) The computer system as claimed in claim 11, wherein the peripheral and expansion ~~bus~~ buses are a PCI bus and an ISA bus, respectively, and the first and second ~~bridge~~ bridges are a north bridge and a south bridge, respectively.

19. (currently amended) The computer system as claimed in claim 11, wherein the second bridge responds to the first data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the first data requests and retrieving the first BIOS code in the ROM corresponding to the addresses.

20. (currently amended) The computer system as claimed in claim 11, wherein the debugging system responds to the second data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the second data requests and retrieving the second BIOS code therein corresponding to the addresses.